Requirements Engineering

Description

Presents best practices for security requirements engineering, including processes that are specific to eliciting, specifying, analyzing, and validating security requirements. Example processes include CLASP, SQUARE, and recent work by Nuseibeh et al. Specific techniques that are relevant to security requirements, such as development of misuse/abuse cases and attack trees and specification techniques such as SCR, are also discussed or referenced.

See also "Threat Modeling: Diving into the Deep End[1]."

Overview Articles

Name	Version Creation Time	Abstract
Security Requirements Engineering	9/30/09 3:22:45 PM	When security requirements are considered at all during the system life cycle, they tend to be general lists of security features such as password protection, firewalls, virus detection tools, and the like. These are, in fact, not security requirements at all but rather implementation mechanisms that are intended to satisfy unstated requirements, such as authenticated access. As a result, security requirements that are specific to the system and that provide for protection of essential services and assets are often neglected. In addition, the attacker perspective is not considered, with the result that security requirements, when they exist, are likely to be incomplete. We believe that a systematic approach to security requirements engineering will help to avoid the problem of generic lists of features and to take into account the attacker perspective. Several approaches to security requirements engineering are described here and references are provided for additional material that can help you ensure that your products effectively meet security requirements.

[1] http://buildsecurityin.us-cert.gov/bsi/resources/articles/932-BSI.html (Threat Modeling: Diving into the Deep End)

Most Recently Updated Articles [Ordered by Last Modified Date]

Name	Version Creation Time	Abstract
Requirements Elicitation Case Studies Using IBIS, JAD, and ARM	3/1/10 3:42:26 PM	This article describes a tradeoff analysis that can be done to select a suitable requirements elicitation method and the results of trying three methods in some case studies. It is a companion to the requirements elicitation introduction[2].
Security Requirements Engineering	9/30/09 3:22:45 PM	When security requirements are considered at all during the system life cycle, they tend to be general lists of security features such as password protection, firewalls, virus detection tools, and the like. These are, in fact, not security requirements at all but rather implementation mechanisms that are intended to satisfy unstated requirements, such as authenticated access. As a result, security requirements that are specific to the system and that provide for protection of essential services and assets are often neglected. In addition, the attacker perspective is not considered, with the result that security requirements, when they exist, are likely to be incomplete. We believe that a systematic approach to security requirements engineering will help to avoid the problem of generic lists of features and to take into account the attacker perspective. Several approaches to security requirements engineering are described here and references are provided for additional material that can help you ensure that your products effectively meet security requirements.
Requirements Elicitation Introduction	2/20/09 10:38:07 AM	Using an elicitation method can help in producing a consistent and complete set of security requirements. However, brainstorming and elicitation methods used for ordinary functional (end-user) requirements

		usually are not oriented toward security requirements and do not result in a consistent and complete set of security requirements. The resulting system is likely to have fewer security exposures when security requirements are elicited in a systematic way. In this article we briefly discuss a number of elicitation methods and the kind of tradeoff analysis that can be done to select a suitable one. Companion case studies can be found in Requirements Elicitation Case Studies[3]. While results may vary from one organization to another, the discussion of our selection process and various methods should be of general use. Requirements elicitation is an active research area, and we expect to see advances in this area in the future. We expect that eventually there will be studies measuring which methods are most effective for eliciting security requirements. At present, however, there is little if any data comparing the effectiveness of different methods for eliciting security requirements.
Requirements Engineering Annotated Bibliography	11/21/08 5:04:48 PM	Abstracts and summaries from the source publications were used for the annotations in this bibliography.
The Common Criteria	11/14/08 3:39:33 PM	The Common Criteria enable an objective evaluation to validate that a particular product or system satisfies a defined set of security requirements. Although the focus of the Common Criteria is evaluation, it presents a standard that should be of interest to those who develop security requirements.

All Articles [Ordered by Title]

Name	Version Creation Time	Abstract
Introduction to the CLASP Process	11/14/08 3:32:49 PM	Comprehensive, Lightweight Application Security Process

		—CLASP—is an activity-driven, role-based set of process components guided by formalized best practices. CLASP is designed to help software development teams build security into the early stages of existing and new-start software development life cycles in a structured, repeatable, and measurable way.
Optimizing Investments in Security Countermeasures: A Practical Tool for Fixed Budgets	11/14/08 3:34:21 PM	Software engineers and businesses must make the difficult decision of how much of their budget to spend on software security mitigation for the applications and networks on which they depend. This article introduces a novel method of optimizing using integer programming (IP), the combination of security countermeasures to implement to maximize system security under fixed resources. The steps in the method and recent results with a case study client are described.
Requirements Elicitation Case Studies Using IBIS, JAD, and ARM	3/1/10 3:42:26 PM	This article describes a tradeoff analysis that can be done to select a suitable requirements elicitation method and the results of trying three methods in some case studies. It is a companion to the requirements elicitation introduction[5].
Requirements Elicitation Introduction	2/20/09 10:38:07 AM	Using an elicitation method can help in producing a consistent and complete set of security requirements. However, brainstorming and elicitation methods used for ordinary functional (end-user) requirements usually are not oriented toward security requirements and do not result in a consistent and complete set of security requirements. The resulting system is likely to have fewer security exposures when security requirements are elicited in a systematic way. In this article we briefly discuss a number of elicitation methods and the kind of tradeoff analysis that can be done to select a suitable

		one. Companion case studies can be found in Requirements Elicitation Case Studies[6]. While results may vary from one organization to another, the discussion of our selection process and various methods should be of general use. Requirements elicitation is an active research area, and we expect to see advances in this area in the future. We expect that eventually there will be studies measuring which methods are most effective for eliciting security requirements. At present, however, there is little if any data comparing the effectiveness of different methods for eliciting security requirements.
Requirements Engineering Annotated Bibliography	11/21/08 5:04:48 PM	Abstracts and summaries from the source publications were used for the annotations in this bibliography.
Requirements Prioritization Case Study Using AHP	11/14/08 3:36:52 PM	This article describes a tradeoff analysis that can be done to select a suitable requirements prioritization method and the results of trying one method, AHP, in a case study. It is a companion article to the requirements prioritization introduction[8].
Requirements Prioritization Introduction	11/14/08 3:37:51 PM	Once you have identified a set of security requirements, you will usually want to prioritize them. Due to time and budget constraints, it can be difficult to implement all requirements that have been elicited for a system. Also, security requirements are often implemented in stages, and prioritization can help to determine which ones should be implemented first. Many organizations pick the lowest cost requirements to implement first, without regard to importance. Others pick the requirements that are easiest to implement, for example by purchasing a COTS solution. These ad hoc approaches are not likely to achieve the security goals of the

		organization or the project. To prioritize security requirements, we recommend a systematic prioritization approach. This article discusses a tradeoff analysis that you can do to select a suitable requirements prioritization method and briefly describes a number of methods. A companion case study [Chung 06][9] can be found in Requirements Prioritization Case Study Using AHP[10]. While results may vary for your organization, the discussion of the various techniques should be of interest. Much work needs to be done before security requirements prioritization is a mature area, but it is one that we must start to address.
Security Requirements Engineering	9/30/09 3:22:45 PM	When security requirements are considered at all during the system life cycle, they tend to be general lists of security features such as password protection, firewalls, virus detection tools, and the like. These are, in fact, not security requirements at all but rather implementation mechanisms that are intended to satisfy unstated requirements, such as authenticated access. As a result, security requirements that are specific to the system and that provide for protection of essential services and assets are often neglected. In addition, the attacker perspective is not considered, with the result that security requirements, when they exist, are likely to be incomplete. We believe that a systematic approach to security requirements engineering will help to avoid the problem of generic lists of features and to take into account the attacker perspective. Several approaches to security requirements engineering are described here and references are provided for additional material that can help you ensure that your

		products effectively meet security requirements.
SQUARE Process	11/14/08 3:38:44 PM	System Quality Requirements Engineering (SQUARE) is a process model that was developed at Carnegie Mellon University, with Nancy Mead as Principal Investigator [Mead 05a].[11] It provides a means for eliciting, categorizing, and prioritizing security requirements for information technology systems and applications. The focus of the model is to build security concepts into the early stages of the development life cycle. The model can also be used for documenting and analyzing the security aspects of fielded systems and for steering future improvements and modifications to those systems.
The Common Criteria	11/14/08 3:39:33 PM	The Common Criteria enable an objective evaluation to validate that a particular product or system satisfies a defined set of security requirements. Although the focus of the Common Criteria is evaluation, it presents a standard that should be of interest to those who develop security requirements.